# INTRODUCTION MARGINAL-WORKERS-CLASSIFIED-AGE-INDUSTRIAL

# **INTRODUCTION.**

This table gives the break-up of the population by their economic activity status as 'main workers', 'marginal workers', 'non-workers' and 'marginal and non-worker' seeking/available for work cross classified with educational level and sex. This table gives the data for India/States/ UTs./Districts and City. This table is separate for SCs upto District level. It allows organizations to make informed decisions related to inventory levels, procurement, pricing, and marketing strategies.

# PREREQUISITES FOR BUILDING A MARGINAL-ORKERS-CLASSIFIED-AGE-INDUSTRIAL-MODEL.

* The data is obtained from https://tn.data.gov.in/catalog/marginal-workers-classified-age-industrial-category-and-sex-census-2011-india-and-states

Have the following libraries installed —

1. Importing OS (data.gov)
2. Numpy and Pandas libraries
3. Matplotlib
4. Seaborn

* Columns Required from dataset
* TABLE CODE
* STATE CODE
* DISTRICT CODE
* AREA NAME
* AGE GROUP

# UNDERSTAND THE SEGMENTATION DATA

Before starting any data science project, it is vital to explore the dataset and understand each variable.

* Libraries Imported :

1. Numpy
2. Pandas
3. Matplotlib
4. Seaborn

* Loading the Data

## df=pd.read\_csv( https://tn.data.gov.in/catalog/marginal-workers-classified-age-industrial-category-and-sex-census-2011-india-and-states)

* let’s look at the head of the dataframe:
* **source code.**

import numpy as np

import pandas pd

import matplotlib.pyplot as plt

import seaborn as sns

from seaborn import load\_dataset

#titanic dataset

data = pd.read\_csv("titanic\_train.csv")

#tips dataset

tips = load\_dataset("

* **Univariate Analysis.**

Univariate analysis is the simplest form of analysis where we explore a single variable. Univariate analysis is performed to describe the data in a better way. we perform Univariate analysis of Numerical and categorical variables differently because plotting uses different plots.

* Categorical Data.

A variable that has text-based information is referred to as categorical variables. let’s look at various plots which we can use for visualizing Categorical data.

* **CountPlot.**

Countplot is basically a count of frequency plot in form of a bar graph. It plots the count of each category in a separate bar. When we use the pandas’ value counts function on any column, It is the same visual form of the value counts function. In our data-target variable is survived and it is categorical so let us plot a countplot of this.

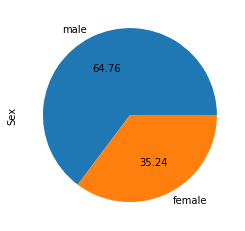
* **Pie Chart.**

The pie chart is also the same as the countplot, only gives you additional information about the percentage presence of each category in data means which category is getting how much weightage in data. let us check about the Sex column, what is a percentage of Male and Female members traveling.

tips

data['Sex'].value\_counts().plot(kind="pie", autopct="%.2f")

plt.show()



# PREPROCESSING DATA FOR SEGMENTATION

The raw data we downloaded is complex and in a format that cannot be easily ingested by customer segmentation models. We need to do some preliminary data preparation to make this data interpretable.

### **Numerical Data,**

Analyzing Numerical data is important because understanding the distribution of variables helps to further process the data. Most of the time you will find much inconsistency with numerical data so do explore numerical variables.

#### Histogram

A histogram is a value distribution plot of numerical columns. It basically creates bins in various ranges in values and plots it where we can visualize how values are distributed. We can have a look where more values lie like in positive, negative, or at the center(mean). Let’s have a look at the Age column.

plt.hist(**data**['Age'], bins=5)

plt.show()

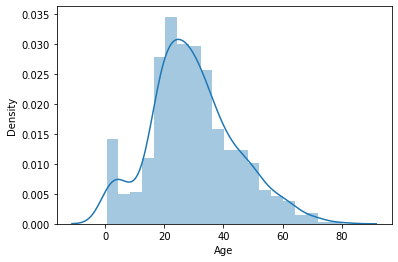


#### Distplot.

Distplot is also known as the second Histogram because it is a slight improvement version of the Histogram. Distplot gives us a KDE(Kernel Density Estimation) over histogram which explains PDF(Probability Density Function) which means what is the probability of each value occurring in this column. If you have study statistics before then definitely you should know about PDF function.

sns.distplot(data['Age'])

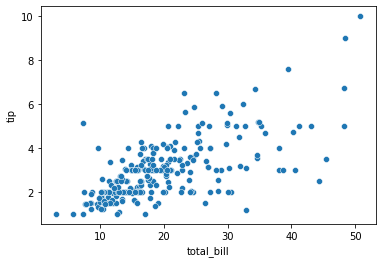
plt.show()



#### Scatter Plot

To plot the relationship between two numerical variables scatter plot is a simple plot to do. Let us see the relationship between the total bill and tip provided using a scatter plot.

sns.scatterplot(tips["total\_bill"], tips["tip"])



* Boxplot

Boxplot is a very interesting plot that basically plots a 5 number summary. to get 5 number summary some terms we need to describe.

* Median – Middle value in series after sorting
* Percentile – Gives any number which is number of values present before this percentile like for example 50 under 25th percentile so it explains total of 50 values are there below 25th percentile
* Minimum and Maximum – These are not minimum and maximum values, rather they describe the lower and upper boundary of standard deviation which is calculated using Interquartile range(IQR).

IQR = Q3 - Q1

Lower\_boundary = Q1 - 1.5 \* IQR

Upper\_bounday = Q3 +  1.5 \* IQR

Here Q1 and Q3 is 1st quantile(25th percentile) and 3rd Quantile(75th percentile)

### Bivariate/ Multivariate Analysis.

We have study about various plots to explore single categorical and numerical data. Bivariate Analysis is used when we have to explore the relationship between 2 different variables and we have to do this because, in the end, our main task is to explore the relationship between variables to build a powerful model. And when we analyze more than 2 variables together then it is known as Multivariate Analysis. we will work on different plots for Bivariate as well on Multivariate Analysis.

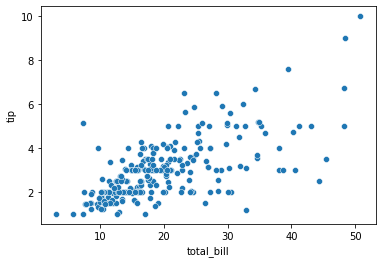
### **Numerical and Numerical.**

First, let’s explore the plots when both the variable is numerical.

#### Scatter Plot.

To plot the relationship between two numerical variables scatter plot is a simple plot to do. Let us see the relationship between the total bill and tip provided using a scatter plot.

sns.scatterplot(tips["total\_bill"], tips["tip"])



#### Multivariate analysis with scatter plot.

we can also plot 3 variable or 4 variable relationships with scatter plot. suppose we want to find the separate ratio of male and female with total bill and tip provided.

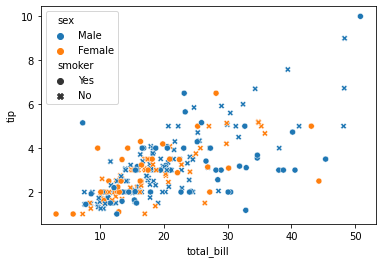
sns.scatterplot(tips["total\_bill"], tips["tip"], hue=tips["sex"])

plt.show()



sns.scatterplot(tips["total\_bill"], tips["tip"], hue=tips["sex"], style=tips['smoker'])

plt.show()

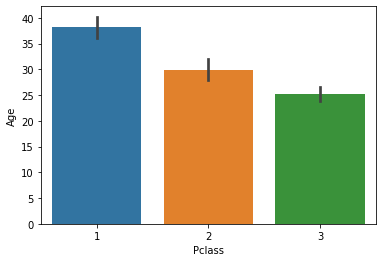


#### Bar Plot.

Bar plot is a simple plot which we can use to plot categorical variable on the x-axis and numerical variable on y-axis and explore the relationship between both variables. The blacktip on top of each bar shows the confidence Interval. let us explore P-Class with age.

sns.barplot(data['Pclass'], data['Age'])

plt.show()

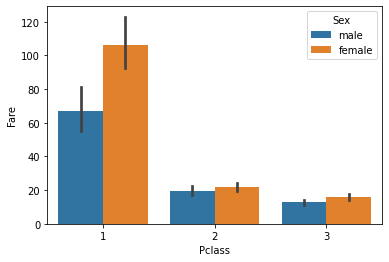


#### Multivariate analysis using Bar plot.

Hue’s argument is very useful which helps to analyze more than 2 variables. Now along with the above relationship we want to see with gender.

sns.barplot(**data**['Pclass'], **data**['Fare'], hue = **data**["Sex"])

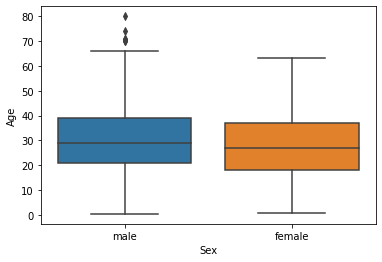
plt.show()



#### Boxplot.

We have already study about boxplots in the Univariate analysis above. we can draw a separate boxplot for both the variable. let us explore gender with age using a boxplot.

sns.boxplot(data['Sex'], data["Age"])

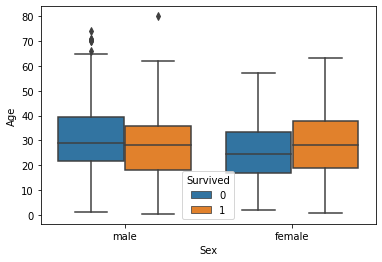


#### Multivariate analysis with boxplot.

Along with age and gender let’s see who has survived and who has not.

sns.boxplot(data['Sex'], data["Age"], data["Survived"])

plt.show()



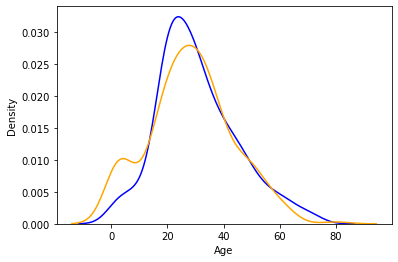
#### Distplot.

Distplot explains the PDF function using kernel density estimation. Distplot does not have a hue parameter but we can create it. suppose we want to see the probability of people with an age range that of survival probability and find out whose survival probability is high to the age range of death ratio.

sns.distplot(**data**[**data**['Survived'] == 0]['Age'], hist=False, color="blue")

sns.distplot(**data**[**data**['Survived'] == 1]['Age'], hist=False, color="orange")

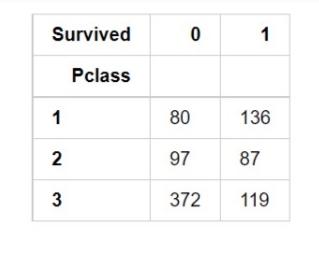
plt.show()



### Heatmap.

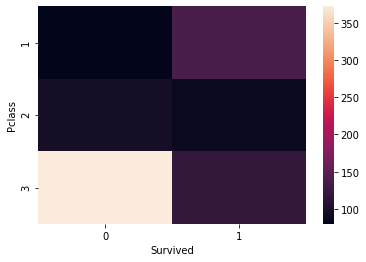
If you have ever used a crosstab function of pandas then Heatmap is a similar visual representation of that only. It basically shows that how much presence of one category concerning another category is present in the dataset. let me show first with crosstab and then with heatmap.

pd.crosstab(data['Pclass'], data['Survived'])



Now with heatmap, we have to find how many people survived and died.

sns.heatmap(pd.crosstab(data['Pclass'], data['Survived']))

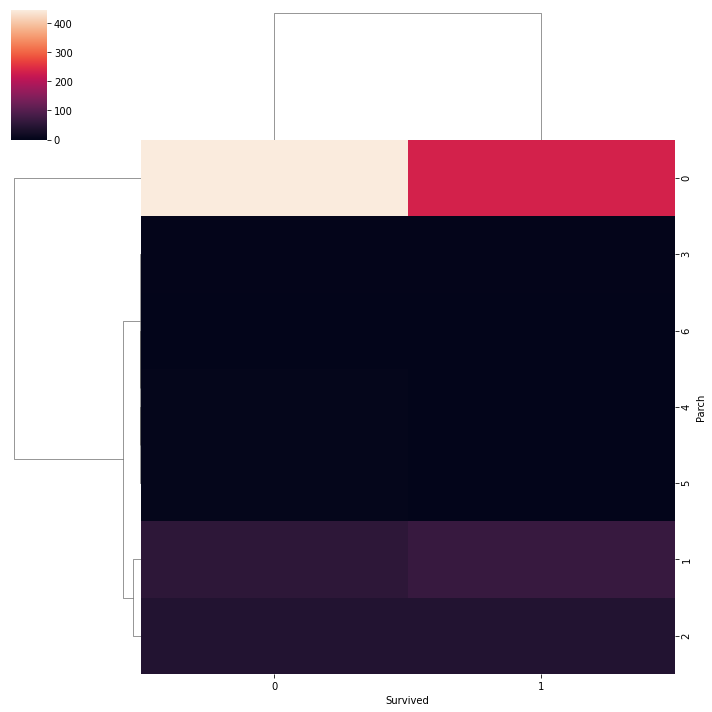


#### Cluster map.

we can also use a cluster map to understand the relationship between two categorical variables. A cluster map basically plots a dendrogram that shows the categories of similar behavior together.

sns.clustermap(pd.crosstab(data['Parch'], data['Survived']))

plt.show()



# TRAIN AND TEST .

Testing the data by importing sklearn.linear\_modal from Linear Regression with ensuring the plot range and axis labels producing the values, scattering the data by mean\_absolute\_error and producing 3D plot. Training the dataset by describe(), isnull().sum(), drop(), show(), and by using Linear Regression algorithm we train the data

Testing the data by importing sklearn.linear\_modal from Linear Regression with ensuring the plot range and axis labels producing the values, scattering the data by mean\_absolute\_error and producing 3D plot.

# REST OF THE EXPLANATIONS.

## Data Collection

The process involves gathering products data, which includes information about their purchase history, demographics, and interaction patterns.

## Data Preprocessing

The task involves preparing and cleaning data, handling missing values, and converting categorical features into numerical representations.

## Feature Engineering

Data preparation and cleaning, handling missing values, and the transformation of categorical features into numerical representations are all part of the task.

Modal Evaluation

Evaluate the model's performance on the test set using appropriate evaluation metrics. Common metrics for demand prediction include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

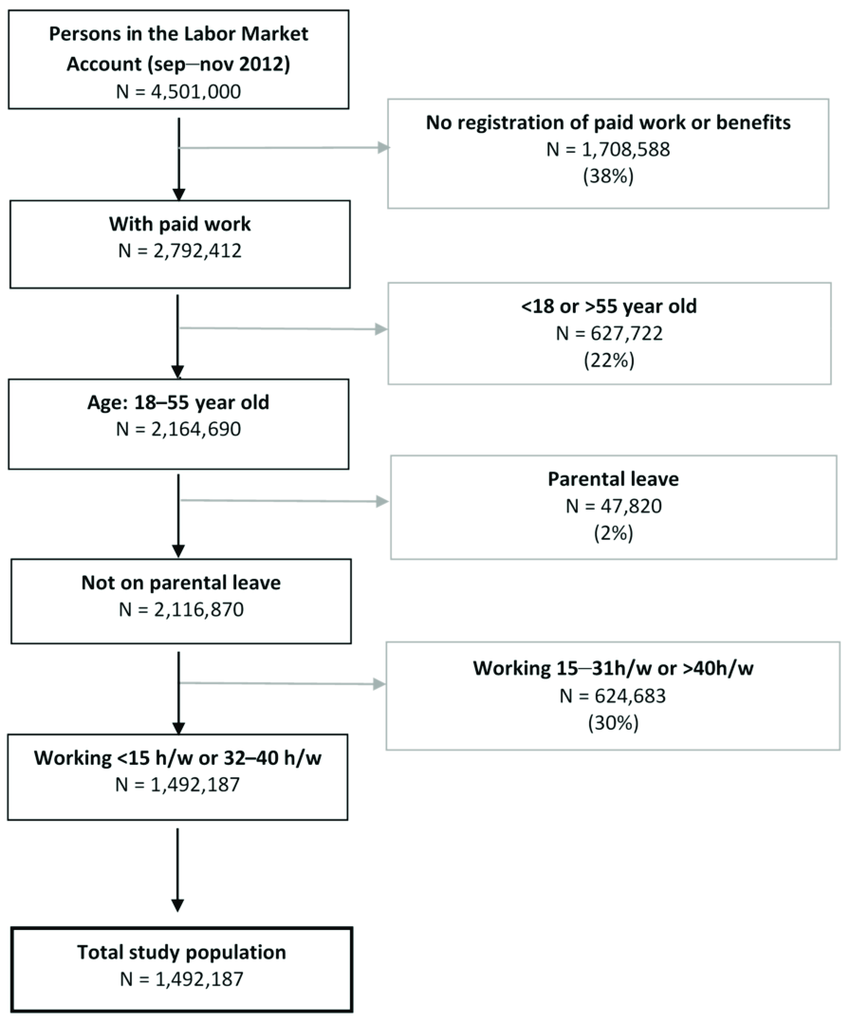
* ALGORITHMS USED .

Apply clustering algorithms like K-Means, DBSCAN, or hierarchical clustering to segment customers.

Visualization: Visualize the customer segments using techniques like scatter plots, bar charts, and heatmaps. Interpretation: Analyze and interpret the characteristics of each customer segment to derive actionable insights for marketing strategies.

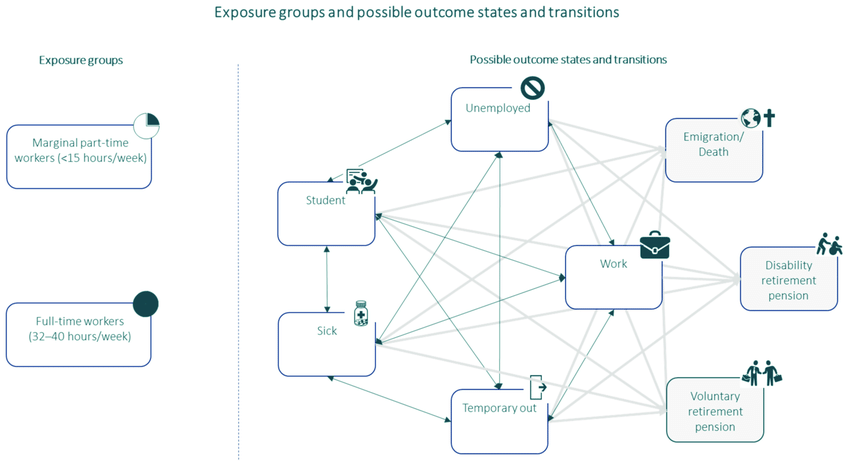
# DATA FLOW OF CUSTOMER MODEL.

* Physical Flow



Figure

* Logical Flow.



THANK YOU